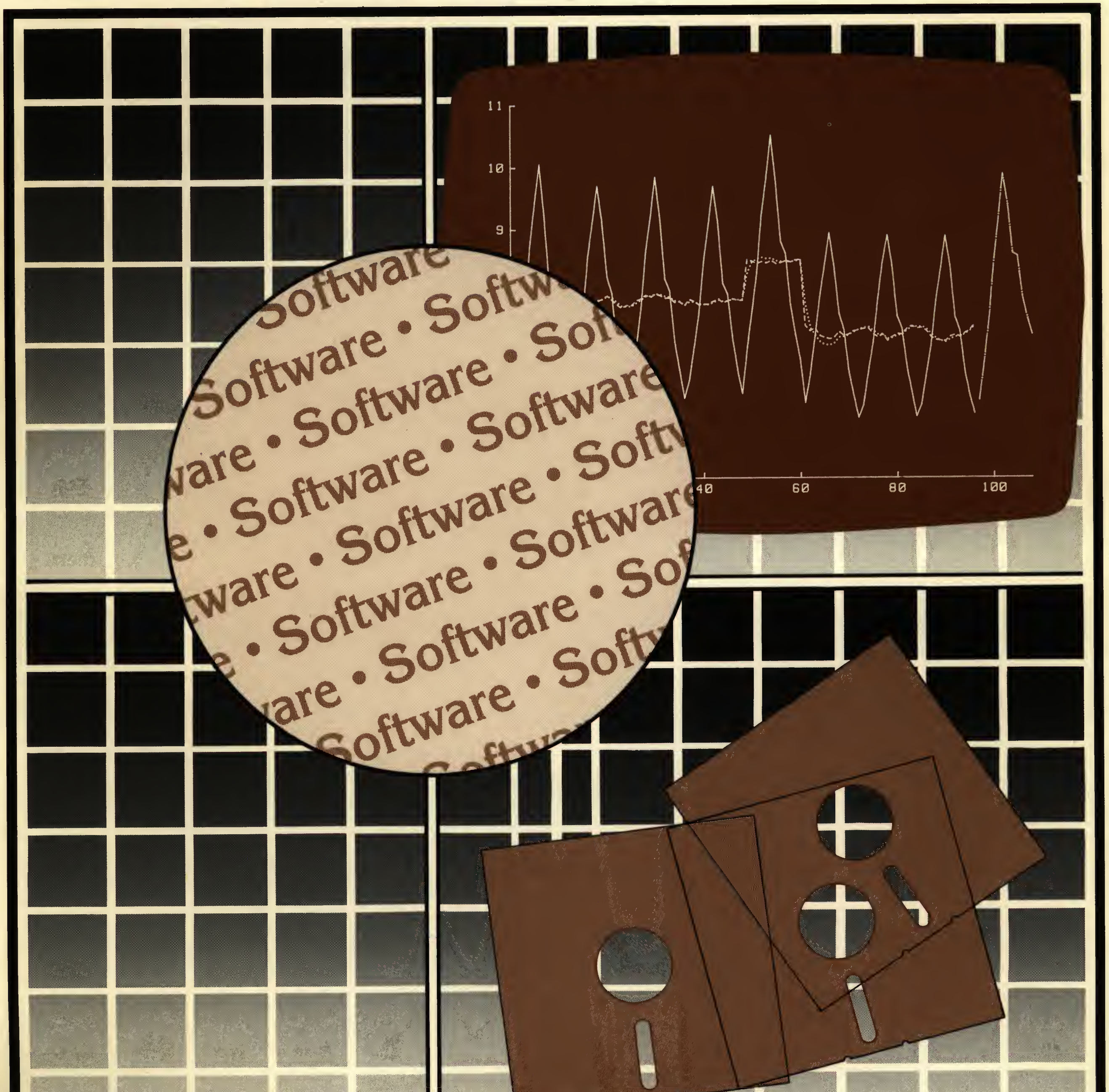
HP 9000 Series 200 Computers

HEWLETT PACKARD

BASIC 3.0 Utilities Library



BASIC 3.0 Utilities Library

for the HP 9000 Series 200 Computers

Manual Part No. 98613-10020

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

May 1984...First Edition

Warranty Statement

Hewlett-Packard products are warranted against defects in materials and workmanship. For Hewlett-Packard Fort Collins Systems Division products sold in the U.S.A. and Canada, this warranty applies for ninety (90) days from the date of delivery.* Hewlett-Packard will, at its option, repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor, parts, and surface travel costs, if any. Equipment returned to Hewlett-Packard for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment, or by hardware, software, or interfacing not provided by Hewlett-Packard are not covered by this warranty.

HP warrants that its software and firmware designated by HP for use with a CPU will execute its programming instructions when properly installed on that CPU. HP does not warrant that the operation of the CPU, software, or firmware will be uninterrupted or error free.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

* For other countries, contact your local Sales and Support Office to determine warranty terms.

Table of Contents

Chapter 1 Orientation

Chapter 2 UTL 3.0 PRG DSC

Chapter 3 UTL 3.0 PRG DSC 2

Orientation

Chapter 1

Introduction

This manual tells you how to use the programs or subprograms contained on two discs labeled: *UTL 3.0 PRG DSC 1*; and *UTL 3.0 PRG DSC 2*. These discs were shipped with your BASIC language system disc (*BSC 3.0 SYS DSC*).

You should not read this manual or attempt to use the utility programs until you become familiar with BASIC 3.0 or unless you are an experienced HP BASIC programmer. You should be able to: configure your system, boot the BASIC language system, load BIN files, initialize discs, and write programs or use applications programs. You should be able to interpret what the computer is doing, or ought to be doing.

This chapter provides general information and descriptions of the utility programs and subprograms. Explanations of how to use the utility programs appear in chapters 2 and 3. Those chapters correspond with the names of the utilities discs: *UTL 3.0 PRG DSC 1* and *UTL 3.0 PRG DSC 2*.

History of Utilities

Typically, utility programs are created when users need a capability not provided by their language system. Thus, they often just accumulate. Documentation is often an assemblage of memos created by programmers, service engineers, or end users.

This has been true to some degree with the HP BASIC utility programs. Consequently, if you are familiar with previous utility programs, you will find that the utilities for BASIC 3.0 differ in small, but significant, ways.

Here are the more significant things to examine if you are a previous user.

a. Some utilities are no longer available. For example, the system management utilities were eliminated because one can now use the BASIC keywords named `XREF` and `SECURE` .

b. Some utilities were consolidated within one utility program. For example, the MASS_STOR utility is a collection of several commonly used separate utilities.

c. Some repartitioning has occurred. For example, the loader utility programs *SYSTEM_LD, CONFIGER*, and *CONFIG_CHK* are on a utilities disc, but the documentation of how they work is a separate manual: *BASIC Loader Utility* manual.

New users should use the utilities as they are presented. Previous users may wish to study the BASIC 3.0 utilities and BASIC 3.0 in relation how they previously used HP BASIC and make appropriate adjustments.

Definitions

A **utility** is a program or subprogram that performs a specific, routine task. Utility programs do such things as: examine the status of an interface, dump a file to a printer, create a backup copy of a file, or display a special list of files. Think of a utility as a program that does some specific, useful task in a manner that goes beyond what you can do with a conventional BASIC statement (e.g. `CAT, COPY, LIST, STATUS`).

A **library** is a collection of programs. You usually think of a library as containing certain types of programs. In this case, you have a library of utility programs.

A **sector** or a **physical record** is a contiguous set of bytes (256 or, perhaps, larger) on a disc that can be accessed.

A **program** is a file of type *PROG*. Utilities that are programs can be loaded and executed.

A **subprogram** can be a *PROG* file, created by using a `SUB...SUBEND` series of statements, or a special *PROG* file that is created by using the *CSUB Utility*. You must create a **main** program that calls and utilizes a subprogram. Your main program (calling program) can often be incorporated into a major program in which you use a utility subprogram.

The **default msus** is the active msus prior to use of a utility. Most utilities use this msus. The *BASIC User's Guide* contains information about mass storage unit specifiers.

A statement in this manual that directs you to **execute** something can mean: press a key, type something and press a key, or load and run a program.

You need to note options for pressing keys or softkeys when you use a utility program. The (Return) key on the HP 46020 keyboard consolidates the (ENTER) and (EXECUTE) keys on the HP 98203A/B keyboards. Some keys on the HP 98203A/B keyboards are equivalent to softkeys in the SYSTEM menu for the HP 46020 keyboard. The HP 46020 keyboard has a different number of softkeys. It has additional menus labeled USER 1, USER 2, and USER 3. If you have an HP 46020 keyboard, most utility program functions are executed by pressing softkeys in the USER 1 or USER 2 menus. The (f8) softkey toggles the menus. Be adept at moving from one menu to another when you have an HP 46020 keyboard. Use the SYSTEM menu to execute commands such as `RUN` and switch to the USER 1 and USER 2 menus to call utility program functions.

System Requirements

A Series 200 computer, monitor, flexible disc drive , and the BASIC language system is the minimal system required to effectively use utility programs or subprograms. Use of some utilities requires additional components such as an interface or printer.

Utilities most often perform particular tasks (in contrast to applications programs which perform a range of tasks within a major task). For example, one utility on disc 2 examines the status of your GPIO interface. It does not do anything else. Having a GPIO interface is a requirement for using the utility effectively. Another utility dumps graphics to a printer. You need a printer to use the utility. Be aware of potential system requirements when you think about how you might use a utility.

Besides hardware, you have to load certain BIN files to use some utilities. An attempt was made to mention required BIN files, but your system may have additional requirements.

CSUBs as Utilities

Most utilities are *PROG* files that can be loaded and executed directly or utilized within a main program that calls a subprogram. You may have a situation where speed is critical.

If you have the programming expertise, you can create a special Compiled SUBprogram that can be accessed via BASIC. Here are the guidelines.

1. Create `CSUB` statements that perform your task using a special CSUB utility from the Pascal language system. The directions for how you do this are in the *CSUB Utility* manual; a package that is acquired separately from BASIC.
2. Create a BASIC program which loads the CSUB subprogram via a `LOADSUB` statement. The subprogram does not have to be loaded within a program, but it is handy to do it this way. After the subprogram is loaded, it can be invoked via a `CALL` statement just like normal SUB subprograms.

Do not think about this too much unless you know what type of utility you need to develop and have considerable programming expertise. If you are curious and wish to learn how to create a CSUB, the process is described in the *CSUB Utility* manual. The *BASIC Language Reference* manual provides additional information.

Why Have a Utilities Library?

Now that you know what this manual contains and how it is organized, you might ask: Why use a utility program? Although it may have been better to ask the question **before** you plowed through the information, here are two illustrations.

Suppose you are working with the HP-IB interface. You have run some programs and are a bit confused about what is happening. You could take a moment to run the HPIB_STAT utility to get a listing of the contents of the status registers for either the internal or external HP-IB interface. This information could help you see what is happening.

Or suppose you have a situation where your discs must meet HP LIF standards (Hewlett-Packard Logical Interface Format) and you are uncertain about the status of several discs. You could use the *VERIFY_LIF* utility to determine whether a disc meets the standard.

These illustrations suggest that utility programs can provide useful information or perform a task easily that would be difficult via conventional methods. Use of utility programs is up to you. Take some time to examine the available programs. Should you decide to use a particular utility, refer to the procedure for using that program.

Available Utility Programs

The utilitiy program are described in the remaining sections in this chapter. Examine the descriptions to learn what the utilities do.

UTL 3.0 PRG DSC 1

The programs and subprograms on this disc are described below. Use the descriptions to guide selection. Go to chapter 2 to learn how to use a particular utility.

Catalog Subprogram (CAT)

This subprogram reads the directory from a disc in the default **msus** into memory as an array. Then, you can access eight pieces of information: file name, file type, starting address, file length, time of creation, volume data, protect code designation, and defined record size. The subprogram also determines the number of files on the disc.

You have to create a program (PROG file) which CALLs the catalog subprogram.

Mass Storage Program (MASS_STOR)

This program is a collection of subprograms. Each subprogram is called by pressing a softkey. The following list is complete, but not in a particular order.

- The Extended CAT subprogram lists an extended directory catalog which includes purged files still in the directory.
- The FILE SIZER subprogram lets you adjust the size of data files.
- The ZAP subprogram gives a used LIF disc the appearance of having just been initialized.
- The Change Volume Label subprogram lets you change the volume label of a LIF disc.
- The PURGE subprogram purges a file.
- The unPURGE subprogram lets you recover from accidental purges.
- The REPACK subprogram packs files on a disc to the front and frees available disc space.
- The Change MSUS subprogram lets you change the current msus.

Use of MASS_STOR varies slightly depending on which keyboard you have.

Formatted Record Dump (DUMP)

This program dumps a specified record from a disc in: integer, hex, octal/ASCII, hex/ASCII, LIF directory, or LIF system formats.

The softkey menus vary slightly depending on which keyboard you have.

Directory Information Subprogram (INFO)

This subprogram accepts a file name. It then searches a disc directory to find a matching file name. Either: the first and last physical record addresses, the number of defined records, the defined record length, and the file type are displayed; or you are informed that no matching file name was found.

You have to create a program that CALLs the utility subprogram.

Create File Subprogram (CREATE)

This subprogram goes beyond the BASIC `CREATE` keyword.

1. You supply a file name, number of defined records, defined record size, and file type.
2. The subprogram creates a file entry in the disc directory and points to where the file's contents can be inserted before, after, or inbetween other files.
3. The new file's address is returned to the calling program if the file's contents will fit on the disc. Otherwise, you are informed that the file will not fit.

You have to create a program that CALLs the subprogram.

Extended Mass Storage Media Initialize (INITIALIZE)

This program lets you specify a legal volume label for an LIF disc. Then, it lets you initialize the disc and specify its directory length.

Verify Logical Interchange Format (VERIFY_LIF)

This program reads record zero on the mass storage media and checks to see whether it meets the HP LIF standards. The status of your media is displayed.

Complete Disc Backup (CBACKUP)

This program, intended for use with one disc drive, lets you backup the contents of a flexible disc which meets HP LIF standards. The disc size can be 3.5, 5.25, or 8 inches.

Do not use with hard discs. Do not use with the `":REMOTE"` msus (SRM).

Selective File Backup (FBACKUP)

When you have one disc drive, use this program to backup particular files on a 3.5, 5.25, or 8 inch flexible disc which meets HP LIF standards.

Do not use with hard discs. Do not use with the `":REMOTE"` msus (SRM). Do not backup the *BSK SYS DSC*. Use *CBACKUP* instead.

CS/80 Tape Backup (TAPEBACKUP)

This program lets you backup the contents of some HP CS/80 discs onto a DC600 or DC150 tape cartridge, and conversely.

The nature of a CS/80 disc drive complicates the procedure somewhat and requires that you coordinate information from several sources. You need to be familiar with your particular CS/80 disc drive and the rest of your computer system. You should coordinate the procedure for using this utility with information about your CS/80 disc and the DC600 tape cartridge.

UTL 3.0 PRG DSC 2

The programs and subprogram on this disc are described below. Use the descriptions to guide your selection of utilities. Go to chapter 3 to learn how to use a particular utility.

List Files (LISTER)

This program provides options for listing BASIC programs.

a. First, it lets you enter the name of one file or enter the names of several files to be listed. The file must have been saved or output as ASCII strings).
b. Then, the values of several parameters are displayed (printer select code, pagination, perforation, lines per page, spacing, omit page numbers, first page number, print range, trailer, edit text, width, number of listings). You can alter these values so that the listing suits your needs.
c. Then, you can do some limited editing of comments.
d. Finally, your program or programs are listed.

A printer is required for effective use. The program is designed to list BASIC programs.

Dump Graphics Subprogram (82905DUMP)

This subprogram accepts a device select code, and then dumps a graphics display to an HP 82905B printer.

You have to create a program that CALLs the subprogram. The subprogram does not work with bit-mapped display.

Creating a Lexical Order Table (LEX_AID)

This program simplifies the process of creating user-defined lexical tables.

You should coordinate the use of this program with information from the string manipulation chapter in the *BASIC Programming Techniques* manual. Do not use this utility unless you know what to do and why you need lexical tables.

Status Utility Programs

Four programs are available which display a formatted listing of the contents of the status registers of either an interface or an ASCII or BDAT file. The programs are:

a. Status of HP-IB Interface (HPIB_STAT)
b. Status of RS232 Interface (RS232_STAT)
c. Status of GPIO Interface (GPIO_STAT)
d. Status of Data Files (FILE_STAT)

These programs display information about what is happening in the registers of an interface or file.

The interface or file must be present for you to get information about the contents of its status registers. Absence of an interface or a file is displayed.

UTL 3.0 PGR DSC 1

Chapter 2

Introduction

Read this section to get basic information about using utility programs.

The *UTL 3.0 PGR DSC 1* disc contains utility programs or subprograms which were described in chapter 1. See that chapter to review what the programs do.

This chapter tells you how to use each utility. Take time to become familiar with HP BASIC before you use utility programs. Work with utilities in a "hands on" setting. Do not just read this chapter.

The special PROG file subprogram named *PHYREC* is discussed at the end of the chapter. PHYREC is used by several utility programs. You need to understand programming and file structure to get much from reading the section.

For each utility program or subprogram, peculiar hardware or BIN file requirements are mentioned. A procedure is provided together with an explanation of what happens. You may need to interpret some directions and make adaptations because, other than having a Series 200 computer, you may have any of several keyboards, monitors, disc drives, or other peripheral devices. You may also have an assortment of interfaces. Your needs or capabilities may not be the same as those of other users.

An attempt was made to provide complete procedures, but you may run into something unusual. Try to work through problems by coordinating information contained in your manuals with what a utility seems to be doing.

The utilities work in a straightforward manner with Model 226 and 236 computers or other Series 200 computers which have an HP 98203A or HP 98203B keyboard. The reason is that many utilities use the 10 softkeys (k0 through k9). If your computer system has an HP 46020 keyboard, orient yourself to beginning with the SYSTEM menu. Then, use the softkeys for the USER 1 and USER 2 menus to call most utility program functions. Softkey f8 lets you toggle between the USER 1 and USER 2 menus.

Remember to configure your BASIC language system so that you can use a utility. Configuration is described in the *BASIC User's Guide*. Configuration becomes more complex as you add external mass storage units, interfaces, and printers to your system. For example, do not attempt to use a utility that requires a printer without first loading the *HPIB* BIN file. Some BIN file requirements are subtle. For example, you need to first load the *PDEV* BIN file to load a subprogram utility via a `LOADSUB FROM` statement. But you can use a `LOADSUB ALL FROM` statement to load a subprogram without first loading the *PDEV* BIN file because this keyword is in mainframe BASIC. Use the *BASIC Language Reference* manual to determine subtle differences in what these keywords do. Examples in this manual use `LOADSUB FROM`. Consequently, *PDEV* is loaded first.

In any event, do not panic. Go slowly. Read carefully. Work through things one step at a time. Stay alert to softkey options as you use a utility. Apply knowledge from a broad base of things you know about using Hewlett-Packard hardware and software to your particular use of a utility.

Note

The utility programs attempt to trap errors, but take care to supply correct entries and press the correct keys. Indiscriminate entries can produce unexpected, detrimental results.

Note

The term **execute** does not mean, press the (EXECUTE) or (EXEC) key. Instead, it means do whatever is implied by the context of a situation. In most cases, you type something and press the (ENTER) or (Return) key. At other times, you execute a command such as `LOAD "MYFILE"`, and then execute another command such as `RUN`.

Note

Consult your HP Sales Representative in situations you cannot resolve.

Catalog Subprogram (CAT)

Skim through the entire procedure before you do anything.

Access this subprogram from within a main program. If you do **not** know how to do this, study the *Subprograms* chapter in the *BASIC Programming Techniques* manual. Fundamental information about storage of data on a media is provided in the*PHYREC* section (end of this chapter).

Do not interrupt the program while it is running. You need a printer to obtain hard copy information. The *PDEV* BIN file is required (if you use `LOADSUB FROM`) in addition to those required to operate your computer system.

The following procedure illustrates **one** way to access the subprogram.

Execute the `SCRATCH` command to ensure that no program is in memory. Then, write a small main program that calls the CAT subprogram. Just how you create the main or calling program can vary. The following steps illustrate the major things to do.

a. Declare parameters to be passed to the subprogram. These are usually input parameters, but can be selected output parameters. `Dstart`, an input parameter, is the address of the sector at which file entries begin. `Dleng`, an input parameter, is the length of the directory. `Num_files`, an output parameter, is the number of files in the directory.

```
10     INTEGER Dstart,Dleng,Num_files
```

b. Initialize parameters that are passed to the subprogram. `Dstart` is set to 2; the usual address for the sector in which file entries begin. `Dleng` is set to 14; the usual directory length of a 5.25 inch flexible disc. Read the *PHYREC* section (end of chapter) to get more complete information.

```
20     Dstart=2
30     Dleng=14
```

The `Num_files` output parameter is initialized in the subprogram, but you should declare it in the main program if you want to output its value after the subprogram has been executed.

c. Dimension the array that receives directory information.

```
40     DIM A$(895)[18]
```

The dimensioning is done according to the expression: `Dleng*64-1`. This is 895 because `Dleng=14` in the above example. The `[18]` specifies a string length of 18.

d. At this point, `CALL` the subprogram. The calling syntax is:

```
50     CALL Cat(A$(*),Dstart,Dleng,Num_files)
```

e. Output information derived from the called subprogram to the CRT, a printer, or a file. This lets you see the information or have it available to use later. For example, the statement:

```
60     PRINT "Number of files is ";Num_files
```

displays the number of files on the disc. Recall that the default print device is the CRT.

With regard to other output, the information read into the array, `A$(*)`, is available in segments of eight returned values per file. The first segment is `A$(0)` through `A$(7)`. The next is `A$(8)` through `A$(15)`, and so on. The eight pieces of information are:

```
1st- File name
2nd- File type
3rd- Starting address
4th- File length in physical records (in defined
     records for BDAT files)
5th- Time of creation (not currently used)
6th- Volume information (not currently used)
7th- Protect code designation
8th- Defined record size
```

The following program segment, inserted in the main program after you call the subprogram, prints file information:

```
70     FOR K=0 TO Num_files*8  ! Print 8 values/file
80        PRINT A$(K)          ! Print array value
90        WAIT .4              ! Let user see values
100     NEXT K                  ! Continue loop
110     PRINT "That's all. You are back in the BASIC system."
```

When you run the program, watch the displayed values. You will see the eight items mentioned above. The 1st value is the file name. The 2nd is the file type and so on. The 8th value is the record size (probably 256).

Printing the values of `A$(*)` this way serves no real purpose other than to let you see what happened. You would probably use the returned values for other purposes.

f. Terminate the context of the main (calling) program. Use:

```
120     END
```

Your main program can do other things, but these statements illustrate a means of calling the CAT subprogram and using data it provides.

Before you run the main program (calling program), insert the disc which contains the CAT subprogram into the default disc drive and load the CAT subprogram into memory by executing:

```
LOADSUB FROM "CAT"
```

This appends the CAT subprogram to your calling program and adjusts the line numbers accordingly. Alternately, you could use:

```
LOADSUB ALL FROM "CAT"
```

To check the appending, execute `EDIT`. Study the program listing to see that the CAT subprogram begins after your main program ended.

At this point, remove the disc which contains the CAT subprogram and insert the disc for which you want data. Then, press the (RUN) key or RUN softkey [SYSTEM menu]. The program accesses your disc and then prints the returned values.

Experiment with other calling programs to develop skill in writing programs that call subprograms.

Mass Storage Subprograms (MASS_STOR)

Skim through the entire section before you use the utility.

Access

Access this program by executing:

```
   LOAD "MASS_STOR"
```

An asterisk, *, is displayed (lower-right corner) while the program loads. Press `RUN` when the asterisk disappears.

General Information

Use of the MASS_STOR utility involves use of nested menus which are displayed via softkey labels. Be aware of when you are invoking a subprogram via a softkey and when you are calling a subprogram function via a softkey. This is especially important when you have an HP 46020 keyboard which requires that you coordinate use of the SYSTEM, USER 1, AND USER 2 menus.

In a similar vein, MASS_STOR is a program that executes subprograms which you call by pressing softkeys. Then, you call subprogram functions by pressing softkeys. The implication is that the softkey labels change often. You need to keep track of what you are doing and which softkey labels are displayed.

Adding to this vein, there is a correspondence between the softkey labels displayed on the screen and the softkeys you press to either: invoke a subprogram, or call a subprogram function. Computer systems which use the HP 98203 A/B keyboards use softkeys k0 through k9. The HP 46020 keyboard uses softkeys f1 through f8 in three menus (SYSTEM, USER 1, USER 2). Note these correspondences as you use the MASS_STOR subprograms because many subprograms and their functions require that you switch menus frequently.

In general, softkey k0, (or f1 [USER 1]), causes the displayed menu and the current **msus** to be printed. Exit MASS_STOR by pressing k9, f7 [USER 1], or f4 [USER 2]. Softkey f8 toggles between USER 1 and USER 2.

Subprogram Utilities

The left column shows the displayed softkey label for a utility provided by the MASS_STOR program. This information is duplicated in a displayed menu when you run the program. The second column indicates:

1. which softkey you press;
2. what the utility does;
3. how you use the utility.

After you invoke a utility, most of its functions are called by pressing a softkey. The labels are self explanatory. The few exceptions are explained. Go slowly and notice alternatives the first time you use a utility.

Softkey Label	Softkey and Procedure
FILE SIZER	k1 or f3 [USER 2]. Lets you change the size of ASCII or BDAT files. When prompted, insert the disc to be sized. Then, enter the filename you wish to change. Abort by pressing ENTER or Return with no entry. Then, you can select: SPECIFY (enter a new file length), FIT DATA (moves end of file to end of last sector with data), or EXIT (returns to main menu). Follow the directions. Do **not** use with `MSI ":REMOTE"` (SRM).
ZAP a disc	k2 or f2 [USER 2]. Clears the directory of a disc by placing a **-1** in the file type field of the first directory entry to denote the logical end of the directory; similar to `INITIALIZE`. When prompted, enter a **msus** (Continue accepts default). Then, insert the correct disc. Then, press the softkey for `YES` to ZAP the disc; or press the softkey for `NO`.
PURGE a file	k3 or f5 [USER 1]. Purges a file on the current **msus**. Just follow the prompts.
UnPURGE a file	k8 or f6 [USER 1]. Unpurges a file on the current **msus**. An extended CAT is displayed. Follow the prompts. Choose the correct type of file (ASCII or BDAT). Note what has and has not been purged during the extended CAT display. Arrows point to files previously purged.
REPACK a disc	k4 or f1 [USER 2]. Repacks a disc to give you a continuous available space at the end of the disc. Pay attention to available softkey options. You are prompted for the **msus**. Type it and press ENTER or Return. Just press ENTER or Return to accept the default. Then, on the prompt, insert the disc to be repacked. Press the CONTINUE key or `Continue` softkey [SYSTEM menu]. Then, note the options via softkeys (USER 1 menu) and proceed. Arrows in the displayed catalog of the directory point to purged files that are lost during repacking.
Change MSUS	k5 or f2 [USER 1]. Lets you specify a new **msus**. The current msus is displayed. Press the softkey for the displayed option you want. Toggle between USER1 and USER 2 to see all options. After selecting an msus, you can alter the select code and address if the displayed code and address are not acceptable.
Extended CAT	k6 or f4 [USER 1]. Displays an extended catalog of a disc. Be aware before you start that eight inch discs require sufficient memory for directory sizes of 28 records and 224 files (single sided) or 58 records and 464 files (double sided).
Chng Vol label	k7 or f3 [USER 1]. Lets you change the volume label of a disc. On the prompt, choose whether you wish to change the current volume label or not. The current label is displayed. If you choose `YES`, enter a new volume label (6 characters, upper-case alpha or numeric, begin with alpha character). If you choose `YES` and then press ENTER or Return with no entry, a volume label containing six blanks is assigned.

Use any of the subprograms above as you require. Work through a subprogram slowly the first time you use it and note options.

Formatted Record Dump (DUMP)

Skim the entire section before you do anything. You need to understand the nature of records and how data is recorded on a disc. You may need to understand LIF. You should be familiar with information contained in the *PYHREC* section (end of chapter). A printer is required.

Access this program by executing:

```
   LOAD "DUMP"
```

When the asterisk disappears, press RUN. Then, insert the disc which contains a record to be dumped and press the (CONTINUE) key or the Continue softkey [SYSTEM menu]. If your disc is **not** in the drive designated by the current **msus**, enter an appropriate msus before you continue.

Several alternatives for dumping a record are selected via softkeys. Toggle between USER 1 and USER 2 to see all options. The default dump is hexadecimal. The options are:

RECORD#	This lets you select the record you wish to dump. The usual range of addresses for records is 0 to 1055. Recall that a 5.25 inch disc has, on two sides, 66 tracks with 16 sectors (physical records) per track. Your situation may vary from this and you must know the range of addresses for records you wish to dump.
Hex	Dumps a record in an hexadecimal array.
Hex/ascii	Dumps a record in an hexadecimal/ASCII format where control characters are masked out and replaced by periods.
LIF sys	Dumps a record in LIF format.
CAT	Displays a catalog of the disc according to the current msus.
N+1, N-1	Allows you to increment or decrement the current record number and dumps a record according to the current format.
Integer	Dumps a record in an integer format.
Oct/ascii	Dumps a record in an octal/ASCII format. Control characters are masked out and replaced by periods.
LIF dir	Causes a subsequent dump to have a LIF directory format.

Examples of dumped records were not provided because use of this utility assumes you know what you are doing.

Note

The program pauses after the fourth entry is printed. Press the (CONTINUE) key or Continue softkey [SYSTEM menu] to print the rest of the record.

Press the EXIT softkey to return to the BASIC system.

Directory Information Subprogram (INFO)

This is another subprogram for which you must create a calling (main) program similar to the one created for calling the CAT subprogram (see CAT Subprogram in an above section). Read through the entire section before you do anything.

Your calling program should do the following:

a. Begin with a `CAT` statement so the directory filename is displayed. Then, input the filename for the directory for which you want information. Use an `INPUT` statement.

```
10    CAT
20    INPUT "Enter file name ",File$
```

b. Declare several parameters as integers. These parameters allow you to pass values to the subprogram or pass values back to the main program (more on this below). Use:

```
30    INTEGER Dstart,Dleng,Strt,Stp,Lrecs,L_rec_l,Ftype,Files
```

c. Initialize the input parameters `Dstart` and `Dleng`. See CAT subprogram section for reasons. Use:

```
40    Dstart=2
50    Dleng=14
```

d. Dimension the array `A$(*)`, which is called when the INFO subprogram calls the CAT subprogram. Again, see Cat subprogram section for reasons. Use:

```
60    DIM A$(895)[18]
```

e. Call the INFO subprogram. Use:

```
70    CALL Info(File$,Dstart,Dleng,Strt,Stp,Lrecs,L_rec_l,Ftype)
```

f. At this point, print as many of the output parameters (described below) as you wish (the ones declared by the `INTEGER` and `DIM` statements above). Include them in your calling program between the `CALL` and `END` statements.

Note that some of the declared variables are output parameters that do not have to be initialized. Including them in the calling statement lets you pass them back to the calling program so you can display their values. The output parameters are:

- Strt An integer variable that contains the address of the first physical record of `File$`. A **-1** means the file was not found. If the file was not found, all other output parameters are **0**.
- Stp An integer variable that contains the address of the last physical record of `File$`.
- Lrecs An integer variable that contains the number of defined records in `File$`.
- L_rec_l An integer variable that contains the length of words in each defined record of `File$`.
- Ftype An integer variable that contains the file type: 1 is ASCII, -5775 is BIN, -5791 is BDAT, and -5808 is PROG.

Note

Be sure the CAT subprogram is on the disc with the INFO subprogram. Load the *PDEV* BIN file if you use a `LOADSUB FROM` statement. Do not interrupt the program while it is actively running.

Notice what is implied by the **Note**. The INFO subprogram calls the CAT subprogram to get additional information. But the disc for which you want directory information may be a different disc. If the CAT subprogram is not on the disc which contains your information, use a `LOADSUB ALL FROM "CAT"` to append the CAT subprogram to your main (calling) program. Then, to make sure the INFO subprogram does not attempt to reload the CAT subprogram, change line `300` in the INFO subprogram to:

```
!  LOADSUB ALL FROM "CAT"
```

The line number will vary if you appended it to your main program, but the `!` keeps the statement from being executed.

Create File Subprogram (CREATE)

This is another subprogram for which you must create a calling program. Apply the procedures used for calling the CAT or INFO subprograms.

The input parameters are:

- File$ Input the name of the file to be created in the directory.
- Num_recs Declare as an integer variable and initialize to the number of defined records in `File$`.
- L_rec_length Declare as an integer variable and initialize to the length of the defined records (words/record for ASCII, PROG, and BIN files; bytes/record for BDAT files). The record length of ASCII, BIN, and PROG files must be 256.
- Ftype Declare as an integer variable and initialize to: 1 for an ASCII file, 0 for an empty file, -1 for a logical end of a directory, -5775 for a BIN file, -5791 for a BDAT file, and -5808 for a PROG file.
- Dstart Declare as an integer variable and initialize to the sector in which the directory starts (probably 2).
- Dleng Declare as an integer variable and initialize to the length of the directory in sectors (probably 14).

The output parameters are:

- New_address An integer variable that returns the file's new address.
- Err An integer variable that returns the status of the create routine: 0 for no valid entry, 1 for no room in directory for entry, 2 for no file space available, or 3 for duplicate file name.

The CALLing syntax is:

```
CALL Create(File$,Num_recs,L_rec_length,Ftype,Dstart,Dleng,New_address,Err)
```

Do not interrupt the subprogram.

Extended Mass Storage Media Initialize (INITIALIZE)

Skim the entire section before you do anything.

Access this program by executing:

```
LOAD "INITIALIZE"
```

Press RUN after the asterisk (lower-right corner) disappears.

On the prompt, enter the **msus** for the device which contains the disc you wish to initialize. ENTER or Return with no entry accepts the current msus. Insert the disc to be initialized and press the CONTINUE key or Continue softkey [SYSTEM menu].

Select the initialization action via softkey. The displayed softkey labels are: YES, NO, or EXIT. Wait while several initialization actions take place. Do not interrupt the process. If you inserted a previously initialized disc, some additional options are provided. They are:

- OKAY (to reinitialize).
- RESTART (to begin the utility again).
- EXIT (to leave the utility).

After initialization, enter a volume name (six characters, uppercase, alpha first character). The default is six blanks. You then enter a directory length and press the CONTINUE key (or Continue softkey [SYSTEM menu]). The default is 14 physical records. You must avoid creating a directory too large to fit onto your disc. Keep directory lengths between 0 and 231. If you selected the NO option, you still have an option to change the volume label within the six character limitations.

The program selects the optimum interleave factor for the given msus.

Logical Interchange Format (VERIFY_LIF)

Skim the entire section before you do anything.

Access this program by executing:

```
   LOAD "VERIFY_LIF"
```

Press RUN when the asterisk (lower-right corner) disappears.

On the prompt, insert the disc to be verified and press the CONTINUE key or the Continue softkey [SYSTEM menu].

Wait a moment. Then, the status of your disc is displayed. It either meets the LIF standard or it does not.

Complete Disc Backup (CBACKUP)

Use this utility when you have one flexible disc drive. Skim through the entire procedure first.

Access this program by executing:

```
   LOAD "CBACKUP"
```

Wait for the asterisk in the lower-right corner to disappear. Press RUN. Wait for the program to go through an initialization process. Do **not** remove the *UTL 3.0 PRG DSC 1* disc until prompted to do so.

Have the master disc (the one containing files you wish to backup) and an initialized backup disc (the one you copy the files onto for backup) available. Do **not** press the (RUN) key or RUN softkey [SYSTEM menu] while the program is backing up a disc. Be patient during apparent inactivity.

Place a PROTECT DATA sticker over the write-protect notch on the master disc if you cannot afford to inadvertently lose data. When prompted, insert the master disc and press the (CONTINUE) key or the Continue softkey [SYSTEM menu]. When prompted, remove the master disc and insert the backup disc and continue.

You can do a CHECK READ, but this is seldom necessary. Copy time is increased considerably. If you decide to do a CHECK READ, the utility reads back what was just written to the disc and compares it with what was just read from the master disc. You may have to exchange discs several times. The exchanges are displayed. On an error, the master disc is read again, the disc is written again, and the two are compared a second time. On a second failure, the error is flagged, a message is displayed, and the program stops. You can PRINT the values of the variables Start and Stop to see the range of addresses within which the error occurred. The CHECK READ and PRINT options are selected by entering Y (yes) or N (no) and pressing (ENTER) or (Return). On completion, you can do another backup or exit. Choose Y or N and press (ENTER) or (Return).

Note

Do not press RUN while the program is backing up a disc. If you do, execute a (PAUSE) or (Shift)(Stop). Then, start over. Do not use this program with the REMOTE msus (SRM).

Selective File Backup (FBACKUP)

Skim the entire procedure first.

Access the program by executing:

```
LOAD "FBACKUP"
```

Press `RUN` after the asterisk (lower-right corner) disappears.

Wait for the program to go through an initialization process. Do **not** remove the *UTL 3.0 PRG DSC 1* disc until prompted to do so.

Put a PROTECT DATA sticker over the write-protect notch of the master disc (the one containing the file you wish to backup). When prompted, remove the utilities disc and insert the master disc. Have an initialized backup disc available. Press the (CONTINUE) key or `Continue` softkey.

The master disc is cataloged. If it is necessary, you can use the cursor keys to scroll the screen. Enter the name of a file you wish to backup and press the (CONTINUE) key or the `Continue` softkey [SYSTEM menu]. Repeat this procedure for each file you wish to backup (limit is 112 files). Then, continue with no typed file entry to begin the copy process.

Note

Do not use this program to backup the *BSC 3.0 SYS DSC.* Use *CBACKUP* instead. Do not use with the `REMOTE` msus. If you inadvertently press `RUN` while the program is active, execute a (PAUSE) or (Shift)(Stop) and start over.

CS/80 Tape Backup (TAPEBACKUP)

Skim the entire section before you do anything. Backing up files stored on a disc in a CS/80 drive which has tape backup capability is more complex than other types of backup. One backup method involves use of switches and buttons found on the disc drive. BASIC provides two methods.

- The TAPEBACKUP utility program provides backup of the contents of a disc.
- The BASIC File System provides backup of selected files.

This section discusses only the TAPEBACKUP utility.

Features and Limitations

Selective file backup is not available strictly within the utility. Some selective file backup is available within the BASIC File System. The TAPEBACKUP utility lets you do a complete backup of an HP 7908, HP 7911, HP 7912, or HP 7914 disc to one or two DC600 tapes; or conversely.

The contents of an HP 7908 disc fit on a DC150 tape. The disc unit number is 0. The tape unit number is 1. The dual controller option is **not** supported. Copying occurs at a rate of 2 Megabytes per minute. The HP 7914 disc requires two DC600 tapes. It can take a few minutes for a tape to load.

Load the *CS80* BIN file if you wish to CATalog the disc or tape. The *HPIB* or *FHPIB* BIN file is required, depending on which interface is installed in your system, but the utility program provides for transfer of data directly from unit 0 to unit 1 without travelling over HPIB.

Access

Access the program by executing:

```
LOAD "TAPEBACKUP"
```

Wait for the * to disappear. Then press (RUN).

Program Options

The utility provides three options: COPY, VERIFY, and CERTIFY. The COPY operation, [USER 1 menu], is called by pressing the (k0) or (f1) key. The VERIFY operation, [USER 1 menu], is called by pressing the (k1) or (f2) key. The CERTIFY operation, [USER 1 menu], is called by pressing the (k2) or (f3) key.

Special Directions

COPY Operation

On the prompt, enter the HP-IB address for the CS/80 disc. Use 702 and press (ENTER) or (Return) for a drive that has select code 7 and address 2. You must know the **msus** for each device you use.

Then, you decide to copy from: the **disc to the tape**, or the **tape to the disc**. This is crucial! Enter the name of the **destination** media. For example, to copy from a disc to a tape, type: TAPE and press (ENTER) or (Return). If you select the tape, you are also asked whether you wish to verify the backup. Select YES now (via softkey) if you wish to verify because an auto-verify checks only that portion of the tape that was written. A stand-alone verify operation checks the entire tape. Selecting YES (to verify) lets you avoid reloading both tapes from an HP 7914 backup to do a stand alone verify.

On the other hand, if you want to copy from a tape to a disc, type: `DISC` and press ENTER or Return.

Since the contents of the destination media are overwritten and lost during the copy operation, here are some particulars you should note:

- Mark the type of drive (e.g. HP 7912) on each tape; especially if you have more than one CS/80 drive.
- Remember that an HP 7914 disc requires two DC600 tapes. Proceed as you would for any other disc; except after the backup with the first tape is completed, remove the tape and insert a fresh one in accordance with the prompt. The backup operation automatically proceeds after the second tape has loaded. For your information, the current `PRINTER IS` device indicates the starting sector numbers for the source and destination units.
- When copying backup contents of two tapes to an HP 7914 disc, the tapes may be loaded in either order because the starting sector number (technically an address) is written on each tape.
- You can perform an HP 7914 backup with one tape when you know that no data is present on the second half of the disc. Do this by pressing the CLR I/O or Break key when you are prompted to insert the second tape.
- Errors are reported on the current `PRINTER IS` device. You can opt to retry the operation or exit.
- When a tape does not verify, do not recertify it. Just repeat the entire copy operation again. The bad blocks detected on the tape are marked and not used again.
- If you copy to an uncertified tape, the tape is automatically certified before the copy begins.
- After recertification, a tape is unloaded. You need to load it again to access it.

Proceed slowly and you should be able to adequately backup your CS/80 discs.

VERIFY Operation

On the prompt, enter the HP-IB address of your CS/80 disc drive (e.g. 702). Then, decide to verify either a tape or a disc. This is crucial! For example, if you wish to verify the tape, type: `TAPE` and press ENTER or Return. If you wish to verify the disc, type: `DISC`.

For safety, you are asked if you wish to proceed. Enter `Y` and press ENTER (or Return) to proceed. Otherwise, enter `N`.

If a disc does not verify correctly, reinitialize the disc. If a tape does not verify, repeat the copy process. Do not recertify it. The bad blocks detected on the tape are automatically marked and not used again. Tapes tend to get better with use.

CERTIFY Operation

Use this operation **only** with tapes. Use this operation when you wish to certify several tapes at one time.

The certify operation works in a manner similar to those described above.

Quitting

Note throughout that you can press the k3 or f4 softkey [USER 1 menu] to `EXIT` from the program.

Some Special Considerations

You cannot access a CS/80 tape via a TRANSFER statement. But you can use the BASIC File System to copy files to a tape, lists a tape's directory, or store programs on a tape.

You can also use BASIC mass storage statements to access a disc image that was copied to a tape via the *TAPEBACKUP* program. This is useful for retrieving a single file from a disc image backup without copying the entire tape back onto the disc. A limitation is that you cannot access a file stored on the second tape used to backup an HP 7914 disc..

Selective retrieval must be done with care due to the tape's slow seek times and inability to start and stop rapidly. Each file written to a tape causes the following tape motion:

1. A seek is performed to the beginning of the tape to scan the directory.
2. The entire directory is scanned one block at a time.
3. A seek is performed to somewhere in the middle of the tape to write the file.
4. A seek is performed back to the beginning of the tape to update the directory.

This can take several minutes due to tape movement. Typical times range from 8 to 60 minutes, depending on which CS/80 drive you have.

Consequently, you should not place very large directories on a tape. Something over 80 entries in a directory is large. Avoid doing unnecessary CAT operations. Avoid copying small files. Avoid file copies when little memory is available.

Remember that two lengths of tapes are available; 17 and 67 Megabytes; DC150 and DC600 respectively. You can use either length.

The *TAPEBACKUP* utility uses some compiled subprograms. These subprograms were generated from compiled code; not from BASIC statements. Consequently, the subprograms cannot be syntaxed within BASIC. This means you cannot modify them via BASIC or access them via a GET statement. You can place a SUB or DEF FN statement directly after the last CSUB statement, but do not use any comments or other BASIC statements.

Physical Record CSUB (PHYREC)

Skim the entire section before you do anything.

PHYREC is a special subprogram called a CSUB which was created by using the Pascal language system *CSUB Utility*. PHYREC lets you do bit-by-bit copies between an integer array in memory and a file on a mass storage media.

The PHYREC program on the *UTL 3.0 PRG DSC 1* disc contains two CSUB statements. They are:

```
10     CSUB Phyread(Sector,INTEGER Int_array(*))
```

and

```
20     CSUB Phywrite(Sector,INTEGER Int_array(*))
```

The Phyread CSUB copies data from the media in the current msus into an integer array. The array can have one to six dimensions. The numeric expression, (e.g. `Sector`), is evaluated to a real and rounded to specify the sector (address) on the disc where the copy begins. The disc is read, beginning with the starting sector, and the data is copied into the integer array in a row-major order. Data is copied until each array element is occupied or until an attempt is made to read beyond the end of data on the media. In the latter case, an error is reported. The implication is that your array should be adequately dimensioned. No pathname is involved. Consequently, the error is trapable by `ON ERROR`, but not by `ON END` statements.

The Phywrite CSUB works the same way, except that data is copied from the array to the media and an error is reported on an attempt to write beyond sector 1055.

Note

The usual `OPTION BASE` circumstances apply to how PHYREC uses arrays. Be sure to use 0 or 1 as required by the nature of your data.

You must load PHYREC into memory before a CSUB is executed (called). Execute:

```
LOADSUB ALL FROM "PHYREC"
```

When PHYREC CSUBs have been loaded into memory, your main program or another subprogram can make calls via `Phyread` and `Phywrite`. One way to see this is to load and list the CAT subprogram. The last two program lines of CAT are:

```
950    CSUB Phyread(Sector,Int_array(*))
960    CSUB Phywrite(Sector,Int_array(*))
```

Other subprogram lines call these subprograms. An example is:

```
100    Phyread((Dstart),A(*))
```

Study the CAT subprogram to conceptualize how PHYREC, CSUB, and Phyread are used. Also, study the Pascal language system *CSUB Utility* and the *BASIC Language Reference* manuals.

Note

Considerable programming savy and computer experience are required to effectively use PHYREC. Do not be put off, but do take time to acquire skill in using HP BASIC.

CAUTION

INDISCRIMINATE USE OF Phywrite CAN CAUSE LOSS OF VALUABLE DATA.

You need to be aware of many things when you use PHYREC. Some considerations, programming hints, and tables follow. The information is not comprehensive. You need to determine how the information is useful.

- The integer array that receives disc data should be dimensioned to minimize disc access. For example, to read the contents of a file composed of 10 physical records (sectors), use a 10 by 128 array with option base 1 where 10 is the number of records and 128 is the size of the words in a physical record.
- You need to understand disc structure to effectively use the PHYREC CSUBs. For example, a 5.25 inch flexible disc contains 66 tracks with 16 addressable sectors per track. Track numbers are not typically used. Thus, the sector addresses range from 0 to 1055. The contents of a sector are important. Its location is not because PHYREC only accesses addressed sectors.
- The contents of a LIF directory on a disc may differ from information displayed when you execute a CAT command because the CAT process interprets some directory information and displays it in a more informative manner.
- A non 5.25 inch disc, initialized by BASIC, probably contains a directory length and file space different from the specifications given in the following *Disc Contents* table. You need to know this information before you use PHYREC.

The following tables contain information about the contents of discs and the differences between directory and CAT values. Use them to conceptualize how information is stored on a disc.

Disc Contents

```
Sector 1, An empty sector

              0  .......................................  0

   Words:               0 through 127

Sectors 2 through 15, The LIF directory sectors

Sector 2    File entry   File entry   File entry  ...  File entry
                1            2            3                8

Sector 3    File entry   File entry   File entry  ...  File entry
                1            2            3                8
   .
   .
   .

Sector 15   File entry   File entry   File entry  ...  File entry
                1            2            3                8
```

Note: Each file entry contains 8 pieces of information stored in words 0 through 15. The 8 pieces of information are:

Respective words used are:

File name	File type	File address	File length	Time of creation	Volume number	Protect code	Defined record size
0-4	5	6-7	8-9	10-12	13	14	15

1) The file name is five words; up to 10 characters.
2) The file type is one word (numeric) as follows:

      1 is ASCII
      0 is Empty
     -1 is Logical end of directory
  -5775 is BIN
  -5791 is BDAT
  -5808 is PROG
  -5822 is SYSTM

3) The file address is two words (numeric); address at which file contents begin.
4) The file length is two words (numeric); numbers of sectors spanned by file's contents.
5) Time of creation is three words set to 0 and not currently used.
6) Volume number is one word; set to -2^15 + 1, not currently used.
7) Protect code is one word; up to 2 characters, 0 for ASCII, defaults to blank for other file types.
8) Defined record size is one word; 0 implies BDAT file with record size of 1 byte, ignored for ASCII but set to 0 for other system compatibility, ignored for PROG or BIN files but set to 128 for compatibility with future products. It may appear on the display as 256.

Note: The directory start address is 2 for 5.25 inch discs. See value of Dstart in procedure for CAT subprogram. The directory length is 14 for 5.25 inch discs. See the Dleng value in the same procedure.

Note: The volume labels for 5.25 inch discs are B9826 or B9836. These can be changed, or can be different for other discs.

Note: The actual content of all files (file space) lies in sectors 16 through 1055.

Differences Between LIF Directory and CAT Values

```
Field                File Type          Directory Value      CAT Value
---------------------------------------------------------------------

file type  .......   ASCII  .................  1  ..........  ASCII
                     Empty entry  ...........  0
                     Logical end of  ........ -1
                         directory
                     BDAT  .................. -5791  .......  BDAT
                     BIN  ................... -5775  .......  BIN
                     PROG  .................. -5808  .......  PROG
                     SYSTEM  ................ -5822  .......  SYSTM

file address  ....   BDAT  ..................  n  ..........  n + 1
                     All others  ............  n  ..........  n

file length  .....   BDAT  ...........  No. sectors  .....  No. sectors
                                                           user sets
                                                           at creation.
                                                           Read words
                                                           4 and 5 of
                                                           system sector.
                                                           Stores as
                                                           first sector
                                                           of file.

protect code .....   ASCII  ................  0
                     SYSTEM  .........  Upper word start
                                        address
                     Others  .........  2 character code

defined record ..    ASCII  ................  0  ..........  256
     size            BIN  ................  128  ..........  256
                     PROG  ...............  128  ..........  256
                     BDAT  ............  If n=0  ..........    1
                                         If n>0  ..........   2n
                                       (length in words) ..  (length in
                                                              bytes)
                     SYSTM  ...........  Lower word
                                         start address
```

Again, it is helpful to learn how to use PHYREC by studying how it is used in other subprograms and by studying the Pascal language system *CSUB Utility* manual.

UTL 3.0 PGR DSC 2

Chapter 3

Introduction

Read this section to get basic information about using utility programs.

The *UTL 3.0 PRG DSC 2* disc contains programs and subprograms described in chapter 1. See that chapter to review what the programs do.

This chapter tells you how to use each utility. Take time to become familiar with HP BASIC before you use the utility programs. Work with utilities in a "hands on" setting. Do not just read the chapter.

For each utility program or subprogram, peculiar hardware or BIN file requirements are mentioned. A procedure for using each utility is provided together with an explanation of what happens. You may need to interpret some directions and make adaptations because, other than having a Series 200 computer, you may have any of several monitors, keyboards, disc drives, or other peripherals. You may have an assortment of interfaces. Your needs or capabilities may not be the same as those of other users.

An attempt was made to provide adequate procedures, but you may encounter something unusual. Try to work through problems by coordinating information contained in your BASIC manual set with what the utility seems to be doing.

The utilities work in a straightforward manner with Model 226 and 236 computers or other Series 200 computers which have an HP 98203A, B keyboard. The reason is that many of the utilities use the 10 softkeys (k0 through k9). If your computer system has an HP 46020 keyboard, orient yourself to beginning with the SYSTEM menu. Then, use the softkeys for the USER 1 and USER 2 menus to call most utility program functions. Softkey f8 lets you toggle between the USER 1 and USER 2 menus.

Remember to configure your BASIC language system so that you can use a utility. Configuration is described in the *BASIC User's* guide. Configuration becomes more complex as you add external mass storage units, interfaces, and printers to your system. For example, do not attempt to use a utility that requires a printer without first loading the *HPIB* BIN file. Some BIN file requirements are subtle. For example, you need to first load the *PDEV* BIN file to load a subprogram utility via a `LOADSUB FROM` statement. But you can use a `LOADSUB ALL FROM` statement to load a subprogram without first loading the *PDEV* BIN file because this keyword is in mainframe BASIC. Use the *BASIC Language Reference* manual to determine subtle differences in what these keywords do. Examples in this manual use `LOADSUB FROM`. Consequently, *PDEV* is loaded first.

In any event, do not panic. Go slowly. Read carefully. Work through things one step at a time. Stay alert to softkey options as you use a utility. Apply knowledge from a broad base of things you know about using Hewlett-Packard hardware and software to your particular use of a utility.

Note

Most utility programs attempt to trap errors, but take care to supply correct entries and press correct keys. Indiscriminate entries can produce unexpected, detrimental results.

Note

The term **execute** does not mean, press the EXECUTE or EXECUTE keys. Instead, it means do whatever is implied by the context of a situation. In most cases, you type something and press the ENTER or Return key. At other times, you execute a command such as `LOAD "MYFILE"`, and then execute another command such as `RUN`.

List Files (LISTER)

If using a printer, load the *HPIB* BIN file. Do **not** use the utility with the `REMOTE` msus. Use of 8.27 by 11.69 inch paper may require alteration of lines 490 and 500 of *LISTER* (see last paragraph in this section)

Access the program by executing:

```
LOAD "LISTER"
```

Wait for the asterisk (lower-right corner) to disappear. Then, press `RUN`.

The program goes through a short initialization process. On the prompt, insert a disc that contains files for which you want program listings. ASCII files created via a `SAVE` statement or `OUTPUT` as ASCII strings are required. Then, press the CONTINUE key or `Continue` softkey [SYSTEM menu] to display a catalog of the disc.

On the prompt, enter a file to be listed and press the CONTINUE key or the `Continue` softkey [SYSTEM MENU]. You can enter up to 112 files. Continuing without entering a file name lets you proceed. All entered files are printed according to the format you establish.

The format for printing a program listing is established via a list of displayed parameters that you can alter. Use the knob to move to a parameter and press the `SELECT` softkey. If you have an HP 46020 keyboard, toggle between the USER 1 and USER 2 menus and press the softkey for a parameter you wish to alter (e.g. press f3 [USER 1 menu] to alter **Paging**). Here are the parameters. They are either: displayed on the screen and accessed via the knob, or displayed on the screen and accessed via softkeys.

`Device Selector` The default is 701. You can enter a different device select code and press ENTER or Return.

`Paging` A toggle parameter (YES or NO). `YES` uses 11 inch pages. `NO` uses continuous printing that breaks between files. Just press the softkey.

`Perforation` A toggle parameter (YES or NO). `YES` assumes perforated (fan-fold) paper. `NO` causes dashes to be printed at points where you should cut the paper. Just press the softkey.

Lines per page

The default is 63 lines. You can alter this to change the top and bottom margins. Just follow the prompts.

Spacing

Another toggle parameter (SINGLE or DOUBLE).

Omit page numbers

Another toggle parameter. The default is NO which retains page numbers. YES omits.

First page number

The default is 1. You can alter this. Just follow the prompts.

Print range

You can select low and high page numbers. ALL prints everything, which means the entire program is listed. ALL can be the high number (e.g. enter 12 as the low number and enter ALL as the high number). Then, the program is listed from the low page number to the end of the program. Execution is slowed when you print a small listing in the middle of a large program because the program goes through the printing process for each line even though only those in the prescribed range are printed. To select ALL, call the function, type ALL. Then press ENTER or Return.

Trailer

You can enter a string which is printed at the bottom of each page next to the page number. DO NOT PHOTOCOPY is an example of a trailer.

Edit text

When set to YES, line numbers and exclamation marks, " ! ", are deleted from the file. For example:

```
10 ! Great ideas
20 ! Are often simple
```

changes to:

```
Great ideas
Are often simple
```

Width

The default is 80 characters. You can change this in the range from 1 to 132.

No. of listings

The default is 1. You can change this as you require. Just remember that beyond 2 or 3, it is much faster to use a copy machine.

EXIT

Exits from the utility.

After you have selected the print format, begin printing by pressing the START PRINTING softkey, (USER 1 menu) on the HP 46020 keyboard. On the prompt, line up the top of form for fanfold paper. Then, press the CONTINUE key or the Continue softkey [SYSTEM menu].

LISTER is programmed to have 70 maximum lines per page. This works with an HP 2631A/G printer. If you have an HP 9876 printer and use 8.5 by 11 inch paper, you have 74 maximum lines per page. If you use 8.27 by 11.69 inch paper, you have 74 maximum lines per page. Thus, if 70 lines per page will not work and you need 74 lines per page, load *LISTER* and EDIT lines 490 and 500 so they read:

```
490     Maxlines=74
500     Ff_space=10
```

Be sure to RE-STORE "LISTER" if you want to make the change permanent.

The 82905B Dump Graphics Subprogram (82905DUMP)

An HP 82905B printer is required. You may need to load the *HPIB, GRAPH,* and *GRAPHX* BIN files. The subprogram does not work with bit-mapped display (Model 237).

You can access this subprogram by creating an appropriate calling program. An example calling program is provided shortly. Remember that a sequence is required. You need to:

1. Load and run a program which contains the graphics you wish to dump to get the graphics image onto the CRT.
2. Load a calling program that can access the dump utility. It is probably better to create the calling program, store it as a PROG file, and then load it when you want to do a graphics dump. Here is the example calling program.

```
10    ! Program to Call 82905DUMP subprogram
20    ! Enter device select code
30    INTEGER Device_selector
40    INPUT "Enter printer select code (e.g. 701)",Device_selector
50    ! Call the subprogram
60    CALL Graphics_dump(Device_selector)
70    ! Reset system defaults
80    PRINTER IS 1
90    GRAPHICS OFF
100   OUTPUT 2 USING "#,K";CHR$(255)&"K"   ! Clear the screen
110   PRINT "The dump is completed.  Proceed as you wish."
120   END
```

3. Append the dump utility to your calling program. Execute:

```
LOADSUB ALL FROM "82905DUMP"
```

You can execute `EDIT` to confirm this.

Now, you should be ready to dump the graphics image. Adjust the paper in the printer so that the left margin is the printer's left margin. Then, run the program. No interaction is required. The graphics display is dumped to the printer in about three minutes. Notice that the last few lines of the calling program reset your system to its original default conditions.

Creating a Lexical Order Table (LEX_AID)

A printer may be required. Some Tables do not fit onto one CRT display. The default lexical order tables for French, German, Spanish, and Swedish are contained in BDAT files on the *UTL 3.0 PRG DSC 2* disc and are labeled *FRENCH, GERMAN, SPANISH,* and *SWEDISH.* Use the LEX_AID program to modify these tables as you require. It is assumed that you know why you need to do this and what you are doing.

There are several stipulations for using the utility.

a. Use the tables in the *String Manipulation* chapter of the *BASIC Programming Techniques* manual to determine the sequence of numbers you assign to each character.

b. When a lexical order table can be developed, scan the table for blocks of consecutive sequence number assignments. The LEX_AID utility has a `FILL BLOCK` function, called via softkey, that helps you assign groups of sequence numbers.

c. Be sure to save the table after you have created it.

d. Invoke a table you have saved with this program.

```
10    INTEGER Table(0:320)
20    ASSIGN @File TO "MYTABLE"  ! Open file
30    ENTER @File;Table(*)       ! Read file
40    ASSIGN @File TO *          ! Close file
50    LEXICAL ORDER IS Table(*)
60    !
70    END
```

Be sure to save the program (`STORE "LEXVOKE"`). In the program, MYTABLE is the name of the file you created with the *LEX_AID* utility. This lexical order holds until you execute `SCRATCH` or another `LEXICAL ORDER IS` statement.

Remember to toggle between USER 1 and USER 2 menus if you have an HP 46020 keyboard. The menu choices, when you run the utility, are:

`Seq Num` This lets you assign a sequence of numbers to a character. Follow the prompts.

`ModeIndex` A submenu is displayed from which mode entries can be selected. The modes are: `Don't Care, 1 for 2, 2 for 1, Accent`, and `Normal.`

`Show Seq` On the prompt, enter a character. Then, the character, its value, and its current sequence numbers are displayed.

`ShowMode` This displays the currently defined mode table entries.

`ShowTable` The mode type and the mode index are displayed as a single value together with current assignments. The information is displayed on the CRT unless you previously redirected output to a printer.

`FillBlock` You are prompted for: the beginning sequence number, and the first and last characters of the block to be filled.

`GetTable` A previously defined lexical order table is loaded from a disc.

`SaveTable` A currently defined table is saved to a disc.

`ListTable` The currently defined lexical order table is printed. The default device select code is 701. You can alter this code.

Again, you should realize that use of this program is based on knowing what you are doing. If you know what to do, load the program and run it. Execute:

```
LOAD "LEX_AID"
```

Press RUN.

To give you some idea, here is an example of using the utility.

1. Load and run the *LEX_AID* utility.
2. Press the FillBlock softkey. Enter 0 for the initial sequence number, 000 for the value of the first character, and 255 for the value of the last character.
3. Press FillBlock again. Enter 65, 097, and 122; respectively.
4. Press the SaveTable softkey. Enter MYTABLE for the filename. Remember that the file will be saved as a BDAT file.
5. Type SCRATCH, and then load, or create, the program that invokes a table (*LEXVOKE* in our previous example). The message, NO DIFFERENCE, is displayed.

This example does not cover really significant usage, but it gives you the idea.

The Status Utilities

Four utility programs are available that let you see the contents of status registers: Data Files (FILE_STAT), HP-IB Interface (HPIB_STAT), RS 232 Interface (RS232_STAT), and GPIO Interface (GPIO_STAT). Each program works in a similar manner. Each requires that you enter fundamental information about such things as device select codes and mass storage unit specifiers.

The best way to learn how to use these programs is to load them and call assorted functions via softkeys. Remember to toggle between the USER 1 and USER 2 menus if you have an HP 46020 keyboard. Be sure to load appropriate BIN files (e.g. load the *GPIO* BIN file when you use the GPIO_STAT utility.

Pay attention to the softkey labels. They are completely self-explanatory. Use then for calling functions. If an interface is not present, press (ENTER) or (Return) with no file entry to exit. Execute any of the four utilities with:

```
   LOAD "FILE_STAT"
```

or

```
   LOAD "HPIB_STAT"
```

or

```
   LOAD "RS232_STAT"
```

or

```
   LOAD "GPIO_STAT"
```

Unlike utilities which can be rather tedious to use, these four are simple and friendly. Yet they can provide valuable information about the contents of Status Registers.

Loader Utilities

The loader utilities consist of three files: *SYSTEM_LD*, *CONFIGER*, and *CONFIG_CHK*. *SYSTEM_LD* is a SYSTM file. The other two are PROG files.

While these three files, which constitute the loader utility, are on the *UTL 3.0 PRG DSK 2* disc, the directions for using them are found in a separate manual: *BASIC Loader Utility*. See that manual for documentation.

Subject Index

Manual Comment Sheet Instruction

If you have any comments or questions regarding this manual, write them on the enclosed comment sheets and place them in the mail. Include page numbers with your comments wherever possible.

If there is a revision number, (found on the Printing History page), include it on the comment sheet. Also include a return address so that we can respond as soon as possible.

The sheets are designed to be folded into thirds along the dotted lines and taped closed. Do not use staples.

Thank you for your time and interest.

MANUAL COMMENT SHEET

BASIC 3.0 Utilities Library
for the HP 9000 Series 200

98613-10020 May 1984

Update No. ________________

(See the Printing History in the front of the manual)

Name: ____

Company: ____

Address: ____

____

Phone No: ______________________

fold -- fold

fold -- fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 37 LOVELAND, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard Company
Fort Collins Systems Division
Attn: Customer Documentation
3404 East Harmony Road
Fort Collins, Colorado 80525

HEWLETT
PACKARD

Part No. 98613-10020
E 0584

Printed in U.S.A.
First Edition, May 1984

HEWLETT PACKARD

Manual Update

This October 1984 update is for:

Loader Utility for the HP 9000 Series 200 Computers
98613-10030
First Edition...May 1984

To update your manual follow the instructions below.

Page ii:
Add "Replaces Loader Utility manual, 09800-10611" and "October 1984 Update" to the Printing History.

Page 5:
Delete the last paragraph on the page.

Add the following paragraph to the bottom of the page:

"A BASIC 3.0 autostart file can be in either the root directory with a file name of AUTOST, or in the SYSTEMS directory with a file name of AUTOSTxx, where xx is the node number. On powerup, the system first looks for an autostart file in the SYSTEMS directory. If it does not find one, it looks in the root directory. Refer to Chapter 2 of the *BASIC Programming Techniques* manual for more information on autostart files."

Fort Collins Systems Division • 3404 E. Harmony Road • Fort Collins, CO 80525

P/N 98613-10030
U 1084
Printed in U.S.A.